

Spatial Joins

1

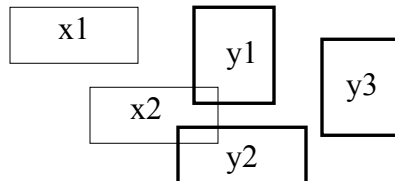
Papers to Present

- “**Efficient Processing of Spatial Joins using R-trees**”, T. Brinkhoff, H-P Kriegel and B. Seeger, Proc. SIGMOD, 1993.
- “**Spatial Joins using Seeded Trees**”, M-L Lo and C. V. Ravishankar, Proc. SIGMOD, 1994.
- “**Spatial Hash-Joins**”, M-L Lo and C. V. Ravishankar, Proc. SIGMOD, 1996.
- “**Size Separation Spatial Joins**”, N. Koudas and K. C. Sevcik, Proc. SIGMOD, 1997.

2

Spatial Join Definition

- Unlike equi-join in relational DBMS, intersection join!



- Join result: $(x2, y1)$, $(x2, y2)$

3

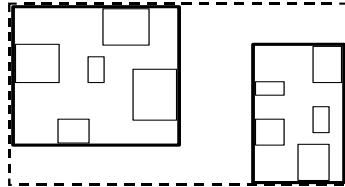
Naïve approach

- For every object in X, check against every object in Y.
- $O(n^2)$ -- not efficient.
- Idea to improve: cluster together objects that are spatially close to each other.

4

R-tree

- Built on each set of spatial objects.



- Objects are clustered into disk pages.
- Recursive cluster till there is one root.
- Benefit: improve on range query.

R-tree based Spatial Join

- Idea: depth-first, synchronized traversal of the two R-trees.
- At the root level: join every child of the first R-tree root with every child of the second R-tree root, if they intersect.

(observation: if two pages do not intersect, no child of page 1 intersect any child of page 2).

- Recursively goes down to leaf level.

Seeded Tree

- An R-tree like structure, built on a set of spatial objects based on an existing R-tree.
- The seeded-tree copies the top levels of the existing R-tree, thus forcing the index to have the same clustering as the existing tree.
- Benefit: improves join efficiency.

Building A Seeded Tree

- **Seeding phase:** copy the top k levels of the existing R-tree. Bottom level contains a set of slots.
- **Growing phase:** insert objects into slots. Each slot may grow to a sub-tree.
- **Clean-up phase:** empty slots are erased; slot MBRs are adjusted.

2. Spatial Join using Seeded Trees

Seeded Tree base Spatial Join

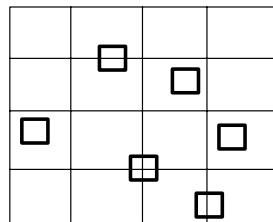
- Used when there exists an R-tree on X, but not on Y.
- Build a seeded tree on Y, then join with the R-tree on X.
- **Seed level filtering:** in the growing phase, if an object does not intersect any seed-level slot, no need to insert!

9

3. Spatial Hash Join

Spatial Hash Join

- Motivation: if all objects fit in memory, then trivial. But size of input is large.
- Basic idea: partition input into smaller buckets that fit in memory.
- Difficulty: an object may intersect multiple partitions!



10

Multiple Assignment Approach

- Partition both input using regular grid.
- Assign an object to all buckets it intersects.
- One bucket in X needs to join with only one bucket in Y.

❖ *Drawback: objects are duplicated, which leads to increase of space.*

Multiple Matching Approach

- Partitioning: start with regular grid, but bucket extent may increase to fully cover all objects assigned to it.
- Assign an object to a single bucket (the one which contains it, or need least enlargement).
- One bucket in X needs to join with all buckets in Y that intersect it.

❖ *Drawback: extensive join of buckets.*

Spatial Hash Join

- Assign every object in X to a single bucket. Bucket extent may increase accordingly.
- Using the same partitioning, assign objects in Y to all buckets that it intersects.
- Join corresponding bucket pairs (a bucket of X joins with a single bucket of Y).

❖ *Better than both previous approaches!*

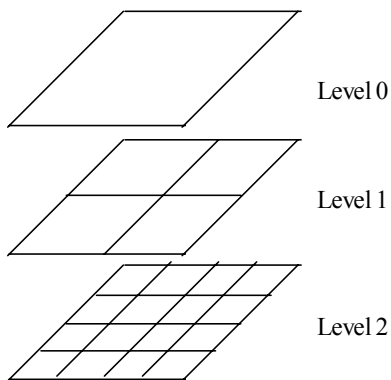
Size Separation Spatial Join Overview

- Partition both object sets by object size into $L+1$ levels.
- Each level is logically partitioned.
- Join the partitions synchronously.

4. Size Separation Spatial Join

Partitioning

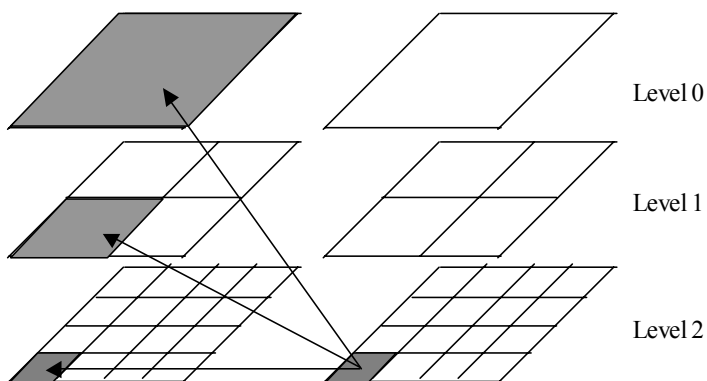
- Level l contains 2^l regular grid cells.
- If an object intersects the grid of level l , assign it to level $l-1$.
- Result: partition (roughly) by size; each object belongs to a single partition at some level.



15

4. Size Separation Spatial Join

Joining



- A bucket in Y needs to join with a single bucket of X at each level.

16

Implementation

- Each level is organized into a file.
- Objects in a level file are sorted by their Hilbert curve value. Each partition corresponds to a Hilbert value range.

Conclusions

- When joining two sets of spatial objects, if both sets are index by R-trees, synchronized R-tree join.
- If only one set has index, build a Seeded Tree for the other relation and join.
- If no index exists, either Spatial Hash Join or Size Separation Spatial Join.